```
# regression_model.py
"""
Boston Housing Price Prediction using Linear Regression

This script trains a multiple linear regression model using the Boston Housing dataset
and evaluates its performance using MSE, R², and Adjusted R².
"""

import numpy as np
import pandas as pd
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Load the cleaned dataset (Ensure 'data/boston_cleaned.csv' exists)
tableData11 = pd.read_csv("../data/boston_cleaned.csv")

# Splitting into features and target variable
np.random.seed(10)
numberRows = len(tableData11)
randomlyShuffledRows = np.random.permutation(numberRows)

# Train-test split (80-20 split)
trainingRows = randomlyShuffledRows[0:405]
testRows = randomlyShuffledRows[405:]

xTrainingData = tableData11.iloc[trainingRows, 0:-1]  # Features
```

```
yTrainingData = tableData11.iloc[trainingRows, -1]    # Target (MEDV)

xTestData = tableData11.iloc[testRows, 0:-1]
yTestData = tableData11.iloc[testRows, -1]

# Train Linear Regression Model
reg = linear_model.LinearRegression()
reg.fit(xTrainingData, yTrainingData)

# Model Coefficients
print("Model Coefficients:", reg.coef_)
print("Intercept:", reg.intercept_)

# Predictions
yPredictions = reg.predict(xTestData)

# Model Evaluation
mse = mean_squared_error(yTestData, yPredictions)
r2 = r2_score(yTestData, yPredictions)

# Adjusted R² Calculation
n = len(xTestData)  # Number of observations
p = len(xTestData.columns)  # Number of predictors
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)

print("\nModel Evaluation Metrics:")
```

```
print(f"Mean Squared Error (MSE): {mse:.3f}")
print(f"R² Score: {r2:.4f}")
print(f"Adjusted R² Score: {adj_r2:.4f}")

# Save results to a file
results = {
    "MSE": mse,
    "R2": r2,
    "Adjusted_R2": adj_r2,
    "Coefficients": reg.coef_.tolist(),
    "Intercept": reg.intercept_
}

results_df = pd.DataFrame([results])
results_df.to_csv("../reports/regression_results.csv", index=False)

print("\nRegression results saved to reports/regression_results.csv")
```